

FIG. 1

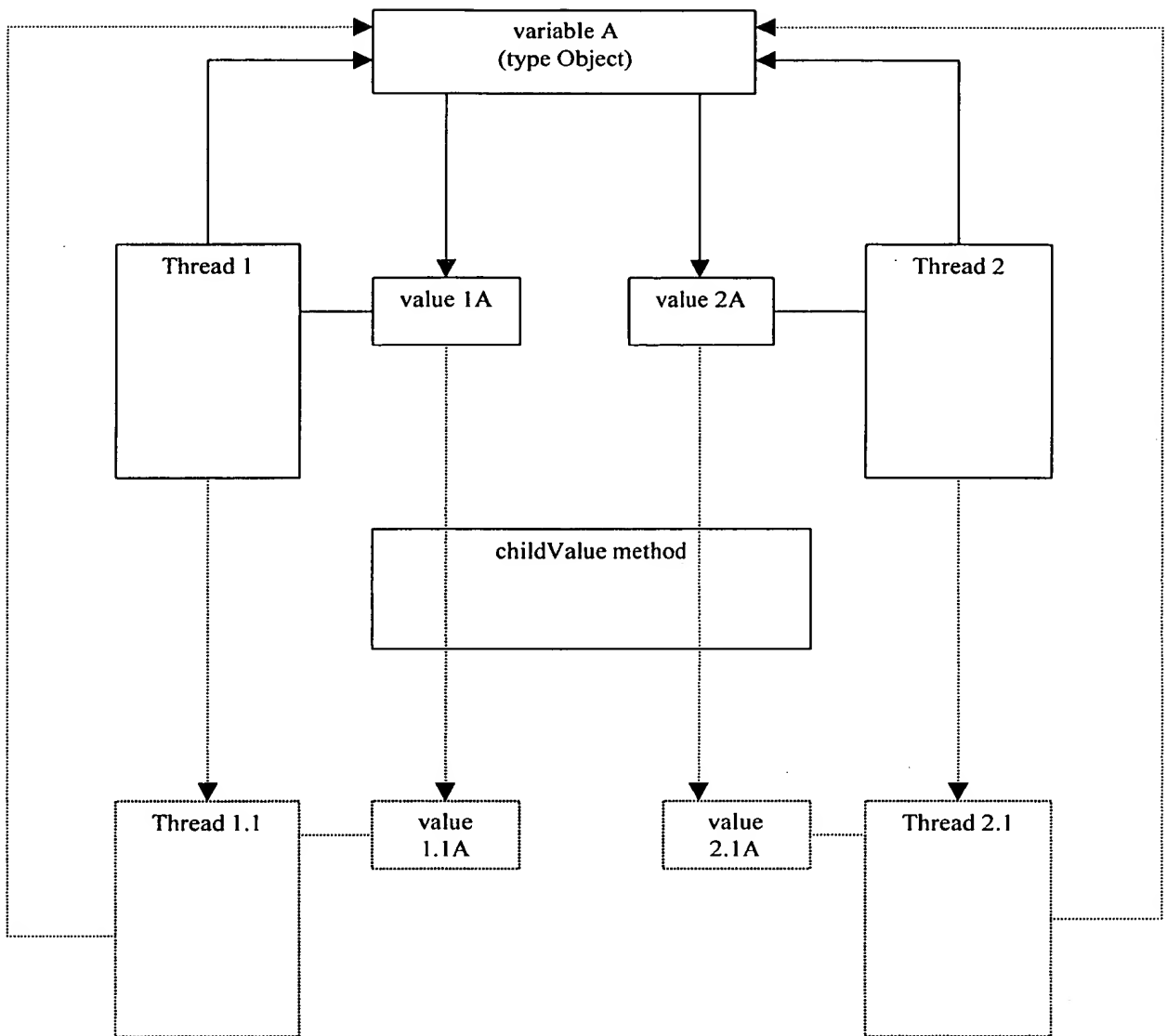


Figure 2

Constructor Summary	
<u>ThreadLocal</u> ()	Creates a ThreadLocal variable.
Method Summary	
<u>Object</u>	<u>get</u> () Returns the value in the calling thread's copy of this ThreadLocal variable.
Protected <u>Object</u>	<u>InitialValue</u> () Returns the calling thread's initial value for this ThreadLocal variable.
<u>void</u>	<u>set</u> (<u>Object</u> value) Sets the calling thread's instance of this ThreadLocal variable to the given value
Methods inherited from class java.lang.Object	
<u>clone</u> , <u>equals</u> , <u>finalize</u> , <u>getClass</u> , <u>hashCode</u> , <u>notify</u> , <u>notifyAll</u> , <u>notifyAll</u> , <u>toString</u> , <u>wait</u> ,	
Constructor Detail	
ThreadLocal	
public ThreadLocal Creates a ThreadLocal variable.	
Method Detail	
initialValue	
protected <u>Object</u> <u>initialValue</u> () Returns the calling thread's initial value for this ThreadLocal variable. This method will be called once per accessing thread for each ThreadLocal, the first time each thread accesses the variable with get or set. If the programmer desires ThreadLocal variables to be initialized to some value other than null, ThreadLocal must be subclassed, and this method overridden. Typically, an anonymous inner class will be used. Typical implementations of initial Value will call an appropriate constructor and return the newly constructed object.	
get	
public <u>Object</u> <u>get</u> () Returns the value in the calling thread's copy of this ThreadLocal variable. Creates and initializes the copy if this is the first time the thread has called this method.	
set	
public void <u>set</u> (<u>Object</u> value) Sets the calling thread's instance of this ThreadLocal variable to the given value. This is only used to change the value from the one assigned by the initialValue method, and many applications will have no need for this functionality. Parameters value - the value to be stored in the calling threads' copy of this ThreadLocal.	

Figure 3 (prior art)

Constructor Summary	
<u>InheritableThreadLocal</u> () Creates an Inheritable ThreadLocal variable.	
Method Summary	
protected <u>Object</u>	<u>childValue</u> (<u>Object</u> parentValue) Computes the child's initial value for this Inheritable ThreadLocal as a function of the parent's value at the time the child Thread is created.
Methods inherited from class java.lang.ThreadLocal	
<u>get</u> , <u>initialValue</u> , <u>set</u>	
Methods inherited from class java.lang.Object	
<u>clone</u> , <u>equals</u> , <u>finalize</u> , <u>getClass</u> , <u>hashCode</u> , <u>notify</u> , <u>notifyAll</u> , <u>toString</u> , <u>wait</u>	
Constructor Detail	
InheritableThreadLocal	
public InheritableThreadLocal () Creates an InheritableThreadLocal variable.	
Method Detail	
childValue	
protected <u>Object</u>	<u>childValue</u> (<u>Object</u> parentValue) Computes the child's initial value for this Inheritable ThreadLocal as a function of the parent's value at the time the child Thread is created. This method is called from within the parent thread before the child is started. This method merely returns its input argument, and should be overridden if a different behavior is desired.

Figure 4

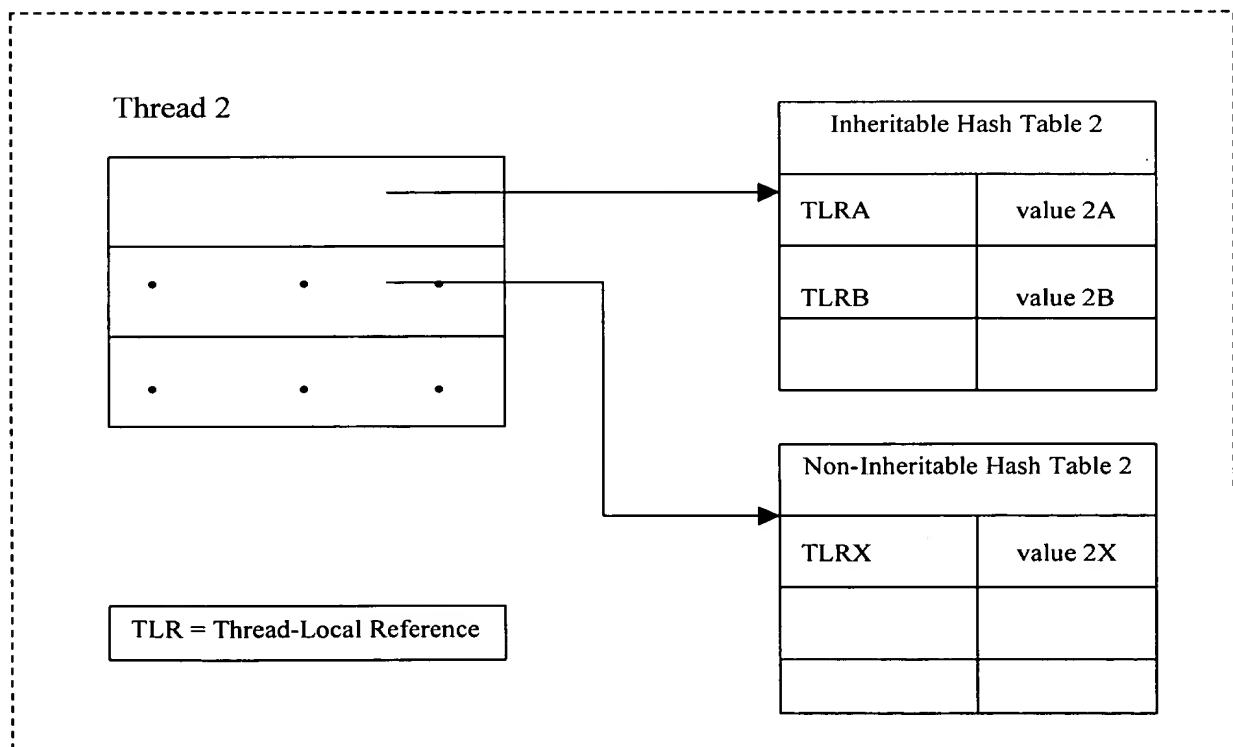
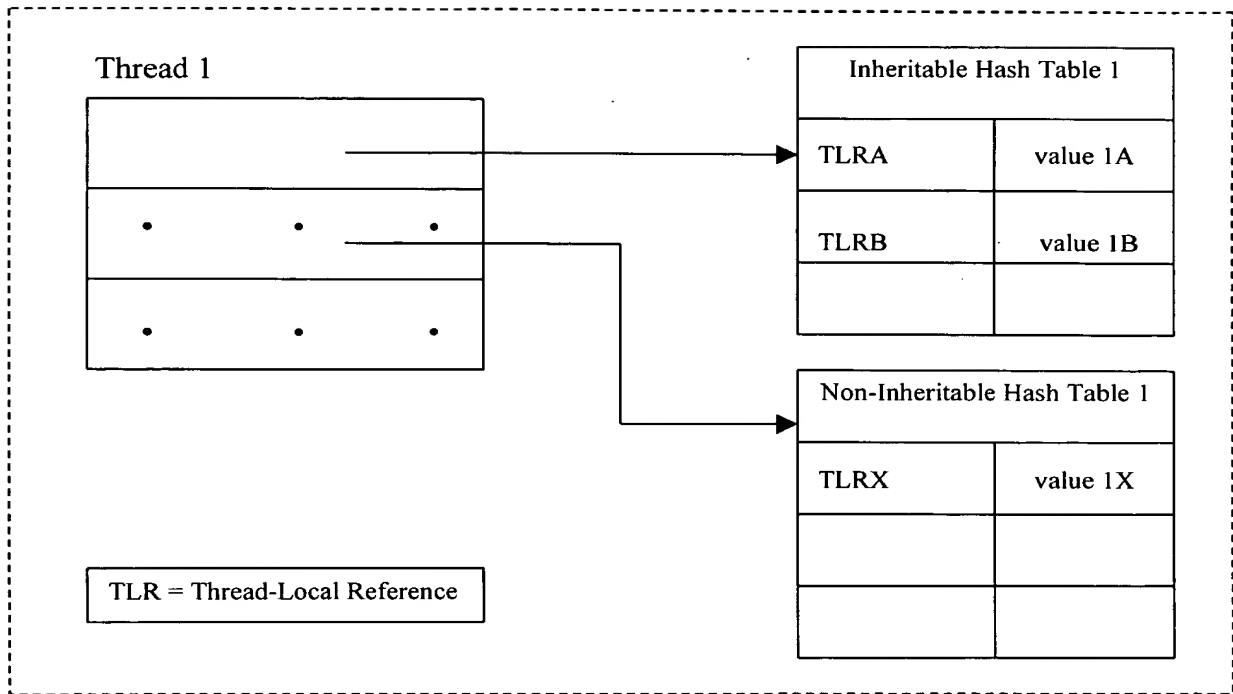
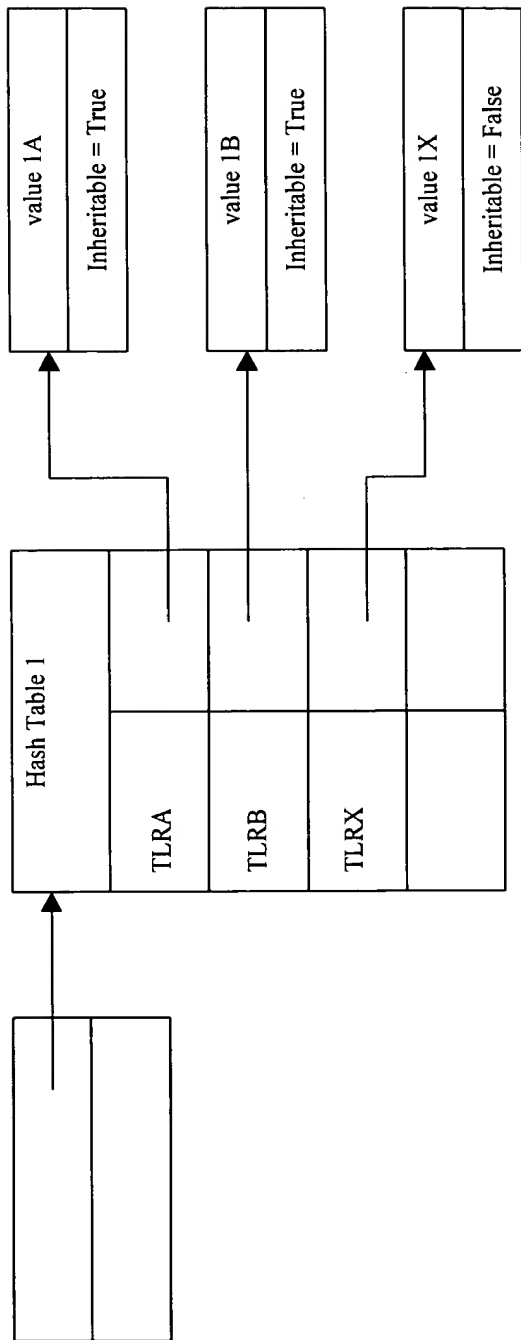


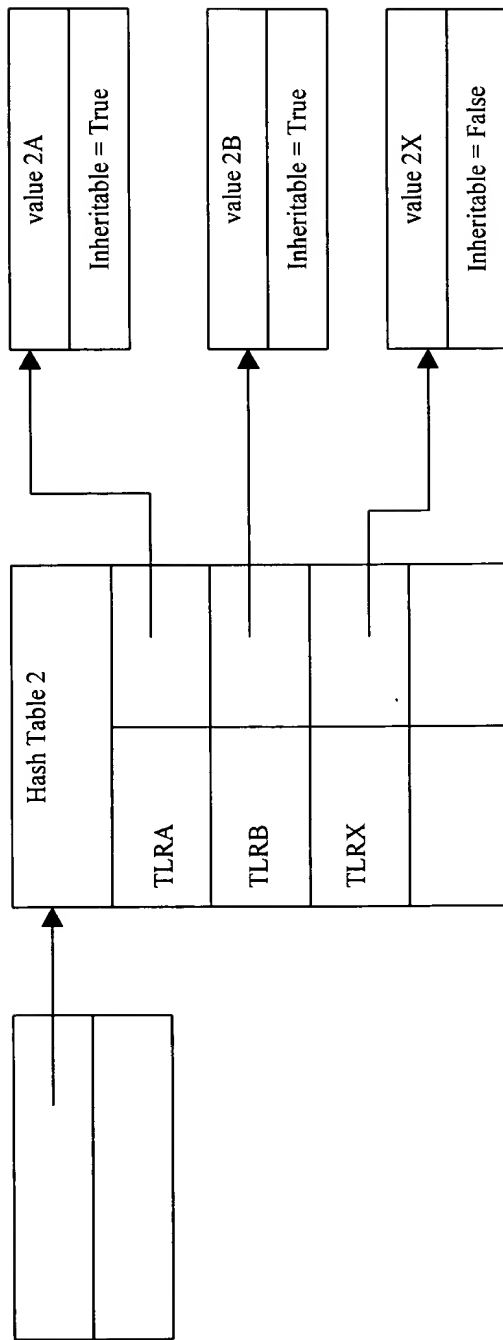
Figure 5

Figure 6

Thread 1



Thread 2



TLR = Thread-Local Reference

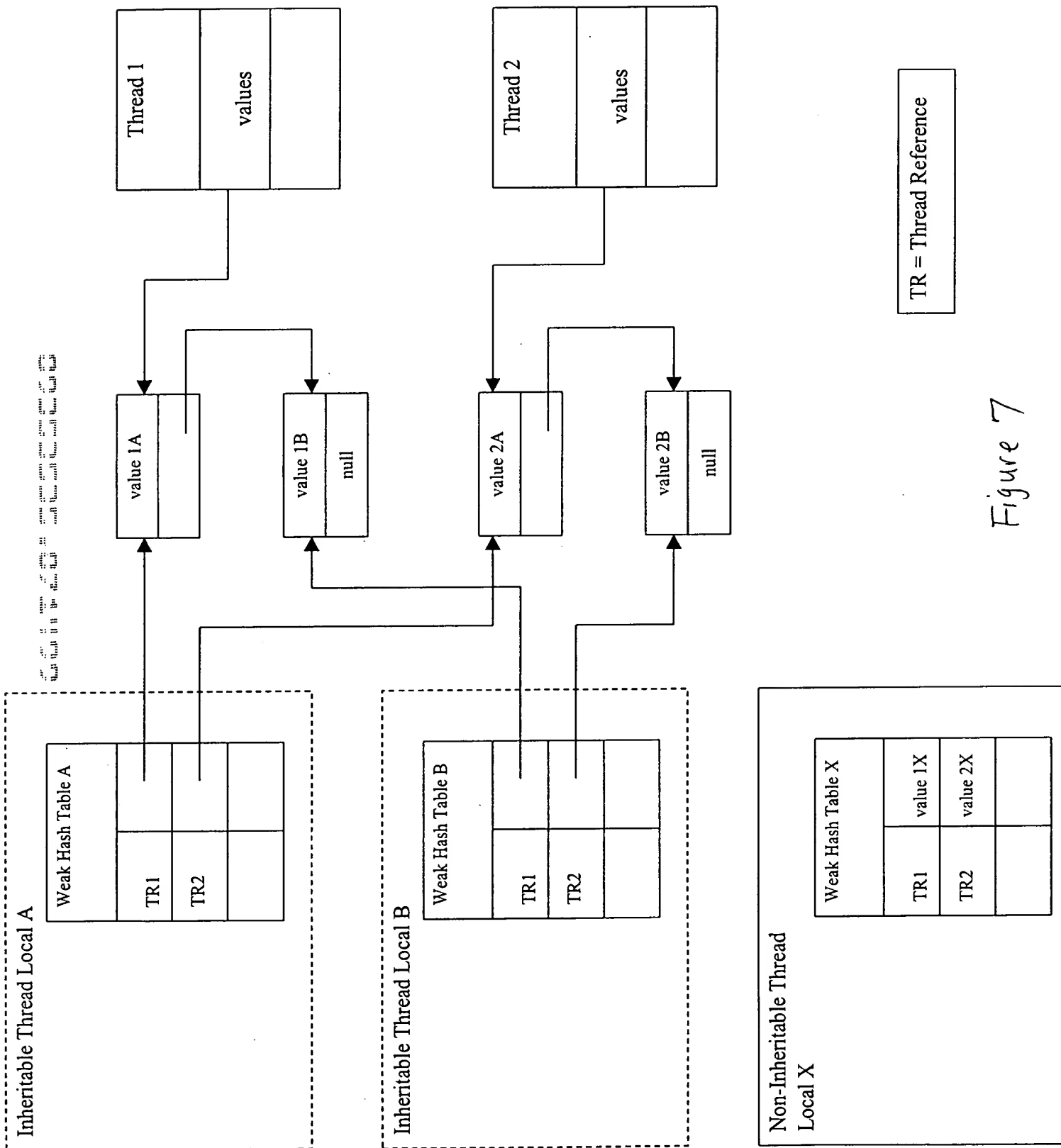


Figure 7